



02

## Centralizando el redireccionamiento

### Transcripción

[00:00] Hola a todos y todas, bienvenidos y bienvenidas a este nuevo video sobre servlet. En este caso, les dije en el video pasado que yo quería que nuestra /entrada servlet sea quien haga obtenga ese JSP y lo envíe al navegador, sin embargo lo que tenemos hoy en día es que nuestra acción es quien está enviando el jsp a nuestro navegador. Entonces vamos a modificar eso.

[00:24] Vamos a ir a nuestro código. Y acá en nuestro servlet, él sería quien haga ese requestDispatcher, él tendría que ejecutarlo, entonces vamos a tomar nuestra lista empresa, por ejemplo acá y vemos que ahí tenemos nuestro requestDispatcher, y no es eso lo que nosotros queremos, entonces vamos a sacarlo de acá.

[00:45] Voy a seleccionar el código, voy a seleccionar hasta acá, "Ctrl + X" y voy a llevarlo a nuestra única /entrada servlet. Lo que uno piensa tal vez en un inicio es hacer esto. Dentro del if si es ("ListaEmpresas") entonces hacer un redirect a "/listaEmpresas.jsp", pero en este caso quiero llevarlo a otro lugar para que vean qué es lo que yo quiero hacer.

[01:10] Dentro de este, voy a sacar acá, en este redirect yo quiero reutilizar este mismo código que sea reutilizado para todas las acciones que requieran hacer un RequestDispatcher, por eso lo coloque acá abajo. Ahora, si yo quiero usar este mismo código para cada una de estas acciones, esta parte del código de acá, este string va a variar dependiendo de cuál acción llame a ese RequestDispatcher.

[01:40] Entonces este string tiene que variar, no podemos dejarlo así harcodeado. ¿Entonces, cómo hago para hacer que ese string varíe? Lo coloco en una variable. ¿Qué es lo que se me ocurrió? Vamos a ver, acá nuestro PPT. Así como nosotros, en nuestro servlet enviamos datos a nuestra acción, estamos enviando nuestro request y nuestro response, nuestra acción también puede enviar datos a nuestro servlet.

[02:10] Aquí tenemos una flecha bien, entonces nuestra acción puede enviar datos a nuestro servlet. ¿Y cuál dato yo quiero enviar? En este primer caso, quiero enviar cuál es el JSP al cual nuestra entrada servlet que tiene que llamar y enviar a nuestro navegador. Entonces vamos a hacer eso.

[02:30] Nuestra acción acá listaEmpresas, en vez de hacer un void, devolver void, no devolver nada, quiero que devuelva un string. ¿Y este string qué es lo que va a contener en este caso? Voy a hacer un return, y este string tiene que devolver un “/listaEmpresas.jsp”, el nombre del archivo jsp al cual nosotros queremos llamar.

[02:59] "Ctrl + S" y entonces si nosotros estamos devolviendo eso de nuestro lado del servlet tiene que recibir ese dato. Entonces vamos a hacer un string nombre = acción.ejecutar. Tendremos nuestro nombre de nuestro jsp, ahora simplemente podemos colocar eso acá en nuestro requestDispatcher y listo, ya tendríamos todo variando, ¿pero qué es lo que ocurre? Está dando un error.

[03:27] Ese nombre no puede ser accedido, si fue creado desde dentro de un if porque si no entra por este if y entra por este otro, por ejemplo, nombre no va a existir, entonces no podría colocarlo acá. Entonces vamos a hacer lo siguiente, en vez de colocar nuestro nombre dentro de nuestra acción if, vamos a sacar la creación de esa variable y vamos a colocarla acá string nombre = null.

[04:07] Voy a colocarle null en un principio, pero se le va a colocar un valor luego dentro de cada if, entonces tenemos ese nombre, ahora sí podemos

llamar a nuestro `requestDispatcher` con ese nombre. Vamos a dar un run a nuestro servidor y vamos a ver si funciona. Bien, ya se reinició, F5, está funcionando.

[04:21] Yo consigo ver ya la consola, es el log, entonces perfecto, eso está funcionando. Vamos a ver ahora el caso de `eliminarEmpresa`, bien vamos a ir a `eliminarEmpresa` y vean lo siguiente, acá no tenemos un `requestDispatcher`, tenemos un `redirect`, entonces vamos a hacer un poco de revisión de nuestros datos.

[04:49] Nosotros teníamos dos formas de enviar una respuesta al navegador, teníamos el `requestDispatcher`, que íbamos directamente a un JSP y ese JSP era enviado al navegador y teníamos la opción de hacer un `redirect` que era decirle al navegador que vaya a otra URL para ahí sí obtener los nuevos datos, el nuevo JSP.

[05:12] Entonces eran dos formas de hacer eso. ¿Ahora cómo vamos a hacer? Yo quiero enviar también `sendRedirect`. Quiero hacer que sea la única entrada servlet quien llame a ese `sendRedirect`, entonces tendría que colocarlo por acá, por ejemplo. ¿Pero ahora cómo sé si mi acción va a llamar a un `requestDispatcher` o va a llamar a un `redirect`? Tengo que tener algo para comparar ahí.

[05:42] Entonces tendríamos que tener algo así como un `if` si ocurre alguna cosa, ya vamos a ver entonces ejecuto mi `redirect`. Pero si no ocurre eso, entonces `else`, hago un `redirect`. Bien. Perfecto. Ahora esa cosa que va a ir acá dentro del `if`, ¿qué es lo que va a hacer? Bueno, una cosa en la que yo pensé, por ejemplo, es en este `listaEmpresas` en vez de enviar solamente mi JSP, quiero enviar también esa condición de qué es lo que tiene que ocurrir si está condición es así.

[06:21] Ya van a ver cuando les escriba el código. Acá quiero, por ejemplo, colocar un `“forward:”`. Entonces, si esta parte de acá a la izquierda de los dos

puntos es forward, significa que yo quiero hacer un requestDispatcher, acá me faltó ahí un espacio, pero si por ejemplo voy a copiar esto de acá, en eliminar empresa yo tengo que hacer esto, tengo que hacer sendRedirect a lista de empresas.

[07:02] Entonces voy a hacer un return y no quiero que sea forward. Forward dijimos que iba a hacer para requestDispatcher, yo quiero que sea un redirect. Entonces coloco redirect. Si es un redirect, no quiero que sean listaEmpresas.jsp, sino que quiero que sea lo que lo que traje para acá, es "entrada?accion=ListaEmpresas".

[07:28] Ahí está. Entonces tanto en lista de empresas como en eliminarEmpresas nosotros tenemos la primera parte, o sea, con primera parte quiero decir a la izquierda de los dos puntos, qué es lo que tengo que hacer, si requestDispatcher o redirect, y la segunda parte es cuál va a ser el camino que voy a usar o el valor digamos que voy a necesitar usar en ese redirect.

[07:51] En el caso del redirect, requestDispatcher, en el caso de requestDispatcher, entonces puede ser que esté un poco confuso, pero ya vamos a verlo en práctica. Acá una cosa que me faltó colocar es que nuestro eliminarEmpresa no retorne un void, sino que sea un string lo que va a devolver. "Ctrl + S".

[08:09] ListaEmpresas, "Ctrl + S". Y acá lo que yo voy a hacer es cómo hago para saber cuál es la primera parte y cuál es la segunda. Si yo solamente uso nombre es difícil realmente saber cuál es la primera parte y cuál es la segunda. Por suerte, Java tiene un método acá llamado split, que lo que nos permite es dividir un string en dos partes y nos crea, gracias a eso nos crea un array con una parte 1 y una parte 2.

[08:45] O sea una parte 0 y una parte 1 en realidad en un array. ¿Entonces, qué es lo que voy a hacer? Voy a hacer un nombre.split y acá yo le tengo que decir cuál es el símbolo dentro de ese string que me divide en dos mi string. En este

caso yo usé unos dos puntos. Esos dos puntos de acá me divide la parte 0 y la parte 1 del string.

[09:12] Punto y coma. Y ahora lo que tengo que hacer es crear un array en el cual voy a guardar esas dos strings. Entonces voy a colocar `string[]` Voy a llamarlo, estuve pensando un poco acá para ver el nombre y ya que tenemos un tipo y una dirección a la cual queremos ir, le voy a llamar al nombre a esta variable `tipoYDirección`. Ese va a ser el nombre, ustedes pueden colocar un nombre mejor.

[09:48] Entonces ese `tipoYDirección` es un array que tiene dos datos, OK en el dato 0 el tipo y en el dato 1 la dirección, entonces voy a preguntar si `tipoYDirección` el dato `[0].equals`, si es igual a "forward", que significa vamos a usarlo para `requestDispatcher`, si ese primer elemento es forward, entonces va a ser un `requestDispatcher`. ¿Pero de qué? No va a ser de nombre, va a ser de la segunda parte de ese `tipoYDirección`.

[10:28] Entonces `tipoYDirección[1]`. Esa segunda parte. Si no, en este caso, significa que es un redirect. Entonces en vez de hacer un redirect directamente de lista de empresas, vamos a hacer un redirect de `tipoYDirección[1]`. Perfecto. Lo último que falta es decirle que este eliminar va a recibir un nombre.

[10:59] Voy a recibir un nombre de ese eliminar y ya con eso estaría listo, porque acá obtenemos nombre, lo dividimos en dos y lo usamos para hacer un `SendRedirect`. Vamos a probar, a ver si esto funciona. Voy a dar un run acá. Vamos a ir a nuestra lista, vamos a ver, nuestra listaEmpresas funcionó. Vamos a ver si eliminar funciona.

[11:23] Eliminar funcionó. Entonces este código está funcionando por ahora. Bueno, ahora lo que falta hacer es todas las otras acciones, `mostrarEmpresa`, `modificarEmpresa`, `nuevaEmpresa`. Les voy a dejar esa tarea a ustedes. Si no consiguen hacerlo, ya vamos a vernos en el próximo video que vamos a realizar esas modificaciones. Entonces nos vemos allá.

